Real Dataset Analysis

```
# Load preprocessed dataset
bladder_comp_adj <- readRDS(here("paper", "data", "bladder_comp_adj.rds"))

# Create a stratified 75/25 split
set.seed(1234)
split <- initial_split(bladder_comp_adj, prop = 0.75, strata = event)

# Create training and testing data frames
train <- training(split)
test  <- testing(split)

# Verify the proportions
table(train$event) / nrow(train)
```

```
#>
#>       0       1       2
#> 0.75111111 0.07111111 0.17777778
```

```
table(test$event) / nrow(test)
```

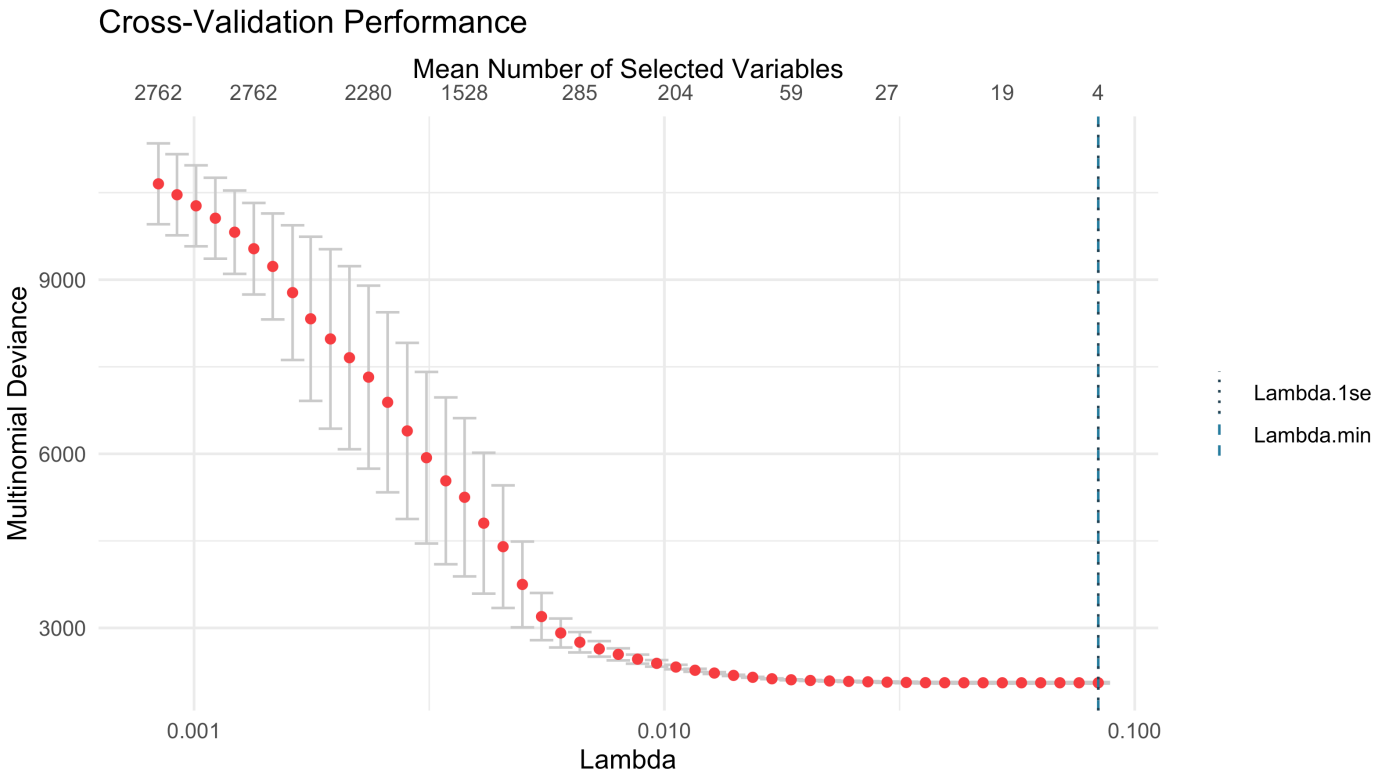
```
#>
#>       0       1       2
#> 0.76315789 0.05263158 0.18421053
```

1 Analysis without Clinical Vars

1.1 cbSCRIP

```
# Set fitting parameters
fit_fun <- purrr::partial(cbSCRIP::MNlogisticAcc,
                          niter_inner_mtplyr = 0.5,
                          maxit = 250,
                          c_factor = 2000,
                          v_factor = 1000,
                          tolerance = 1e-3,
                          save_history = F,
                          verbose = F)

if(save){
```

```
  set.seed(123)

  cv_nc <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    train[,-(3:7), , drop = FALSE],
    alpha = 0.7,
    nfold = 5,
    nlambda = 50,
    fit_fun = fit_fun,
    ratio = 50)

  plot(cv_nc)

  saveRDS(cv_nc, here("paper",
          "results",
          glue("cv_nc.rds")))
}

cv_nc <- readRDS(here("paper", "results", glue("cv_nc.rds")))

# Print c-plot
plot(cv_nc)
```




```
# Print selected vars
## filter rows
filt_rows <- which(!same(cv_nc$fit.min$coefficients[,1], 0) |
      !same(cv_nc$fit.min$coefficients[,2], 0))
```

```
cv_nc$fit.min$coefficients[filt_rows,]
```

```
#>              [,1]     [,2]
#> log(time)  -0.4213303 -0.5532271
#> (Intercept) -4.9795840 -4.0324360
```

```
# Fit model with Lambda(min + 1SE)
lambda_min_nc <- cv_nc$lambda.min
lambda_min_nc_idx <- which(lambda_min_nc == cv_nc$lambdagrid)
dev_lambda_nc <- cv_nc$deviance_mean[lambda_min_nc_idx]
se_lambda_nc <- cv_nc$deviance_se[lambda_min_nc_idx]
lambda_min_minus_se_idx <- which.min(abs(cv_nc$deviance_mean - (dev_lambda_nc + se_lambda_n
lambda_min_minus_se <- cv_nc$lambdagrid[lambda_min_minus_se_idx]

# Fit model with Lambda(min + 1SE)
p1se_nc <- cbSCRIP(
     cb_data = cv_nc$cb_data, # using case-base sample generated by cv function
     alpha = 0.7,
     lambda = lambda_min_minus_se,
     fit_fun = fit_fun,
     ratio = 50)

# Print selected vars
filt_rows_p1se <- which(!same(p1se_nc$coefficients[[1]][,1], 0) |
      !same(p1se_nc$coefficients[[1]][,2], 0))

p1se_nc$coefficients[[1]][filt_rows_p1se,]
```

```
#>              [,1]     [,2]
#> seq1082    0.0000000  0.00000001
#> seq1225    0.0000000 -0.09488445
#> seq1226    0.0000000 -0.05371728
#> seq240     0.0000000 -0.05967735
#> seq249     0.0000000 -0.08177938
#> seq265     0.0000000  0.00000007
#> seq279     0.0000000 -0.01524196
#> seq302     0.0000000 -0.08624647
#> seq336     0.0000000  0.07910201
#> seq339     0.0000000  0.01074220
#> seq34      0.0000000  0.23578966
#> seq377     0.0000000  0.10402633
#> seq435     0.0000000  0.00000043
#> seq813     0.0000000 -0.12108261
#> seq833     0.0000000  0.13213085
#> seq869     0.0000000 -0.00000001
#> seq972     0.0000000  0.06828371
#> seq973     0.0000000  0.08746388
#> seq982     0.0000000  0.15114483
#> log(time)  -0.6678312 -0.69929120
```

```
#> (Intercept) -4.1964660 -3.05984924
```

```
# Repear refitting multiple times

if(save){

   set.seed(123)

   for (i in 1:50) {
   p1se_nc <- cbSCRIP(
      Surv(time, event) ~ .,
      train[,-(3:7), , drop = FALSE],
      alpha = 0.7,
      lambda = lambda_min_minus_se,
      fit_fun = fit_fun,
      ratio = 50)

   if(i== 1) {count_mtx <- !same(p1se_nc$coefficients[[1]],0)
   } else {
      count_mtx_loop <- !same(p1se_nc$coefficients[[1]],0)
      count_mtx_loop[] <- as.integer(!same(p1se_nc$coefficients[[1]],0))
      count_mtx[] <- count_mtx + count_mtx_loop
   }

   }

   saveRDS(count_mtx, here("paper",
           "results",
           glue("count_mtx.rds")))
}

count_mtx <- readRDS(here("paper", "results", glue("count_mtx.rds")))

filt_rows <- which(!same(count_mtx[,1], 0) |
      !same(count_mtx[,2], 0))

# Number of variables selected out of 50
count_mtx[filt_rows,]
```

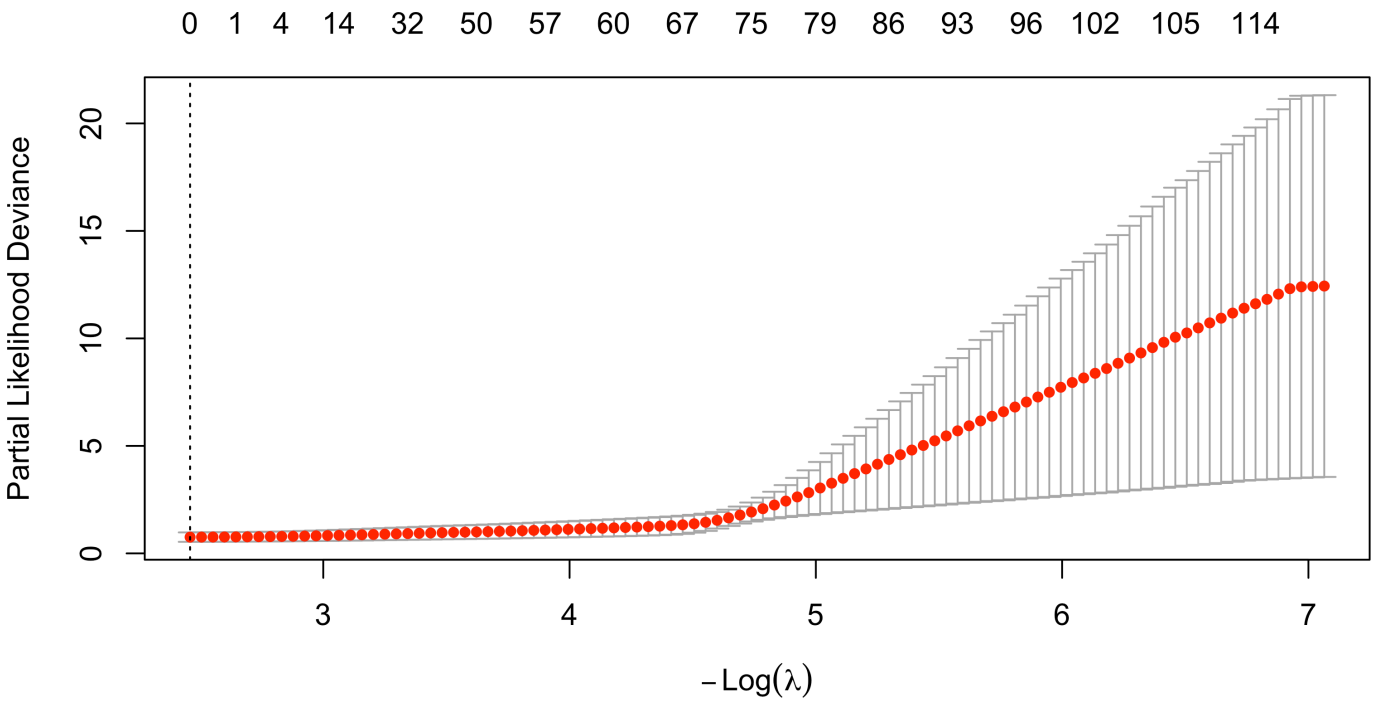
```
#>         [,1] [,2]
#> seq1014    0   3
#> seq1029    0   1
#> seq1031    0   1
#> seq1082    0   8
#> seq1145    0   1
#> seq1177    0  29
#> seq1225    0   9
#> seq1226    0  12
#> seq1227    0   1
```

```
#> seq1262     0  12
#> seq1310     0   4
#> seq1364     0   1
#> seq1384_2    0  33
#> seq227      0   1
#> seq240      0  40
#> seq249      0  13
#> seq265      0  50
#> seq279      0  36
#> seq324      0   6
#> seq336      0  18
#> seq339      0  40
#> seq34       0  50
#> seq370      0  24
#> seq377      0  50
#> seq408      0   3
#> seq435      0  14
#> seq438      0   1
#> seq440      0  10
#> seq445      0   2
#> seq49       0   1
#> seq542      0   1
#> seq62       0   1
#> seq681      0   1
#> seq78       0   2
#> seq782      0   2
#> seq793      0   1
#> seq813      0  44
#> seq820      0  11
#> seq833      0  34
#> seq866      0   1
#> seq869      0  10
#> seq919      0   1
#> seq940      0   3
#> seq963      0  13
#> seq972      0  48
#> seq973      0  41
#> seq982      0  31
#> log(time)   50  50
#> (Intercept)  50  50
```

1.2 Cox elastic-net

```
y <- Surv(time = train$time,
       event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
            data = train[,-(3:7), , drop = FALSE])
```

```r
set.seed(1234)
cox_enet_nc <- cv.glmnet(x = x, y = y, family = "cox",
                # family = "binomial",
                nfolds = 5,
                alpha = 0.7)

plot(cox_enet_nc)
```




```r
# Print selection
cc_enet_min_nc <- coef(cox_enet_nc, s = cox_enet_nc$lambda.min)

select_vars_enet_nc <- cc_enet_min_nc@Dimnames[[1]][-1][cc_enet_min_nc@i]

selected_coefs_enet_nc <- cc_enet_min_nc@x

names(selected_coefs_enet_nc) <- select_vars_enet_nc

selected_coefs_enet_nc
```

```
#> named numeric(0)
```

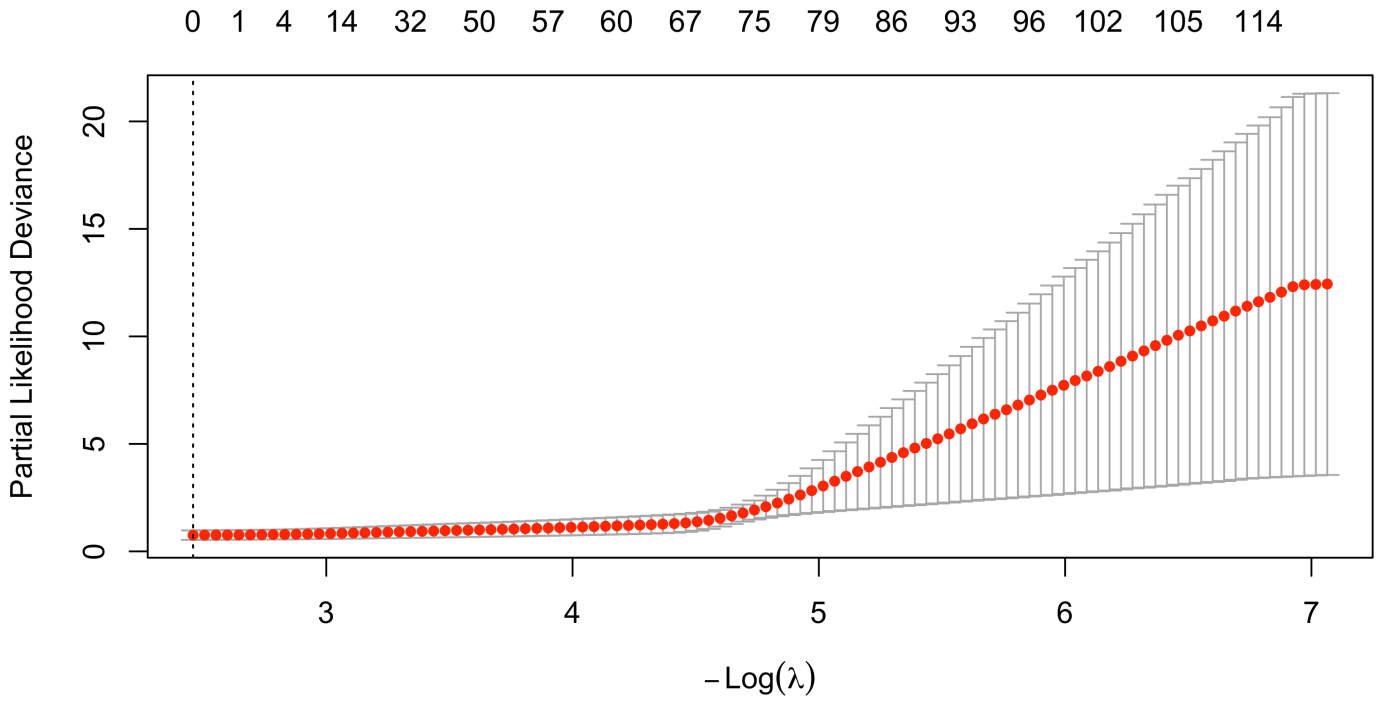
```r
# Model w 1SE below

cox_lambda_min_nc <- cox_enet_nc$lambda.min
cox_lambda_min_nc_idx <- which(cox_lambda_min_nc == cox_enet_nc$lambda)
dev_cox_lambda_nc <- cox_enet_nc$cvm[cox_lambda_min_nc_idx]
se_cox_lambda_nc <- cox_enet_nc$cvsd[cox_lambda_min_nc_idx]
cox_lambda_min_minus_se_idx <- which.min(abs(cox_enet_nc$cvm - (dev_cox_lambda_nc + se_cox_
cox_lambda_min_minus_se <- cox_enet_nc$lambda[cox_lambda_min_minus_se_idx]
```

```r
cc_enet_min_minus_se_nc <- coef(cox_enet_nc, s = cox_lambda_min_minus_se)

select_vars_enet_nc <- cc_enet_min_minus_se_nc@Dimnames[[1]][-1][cc_enet_min_minus_se_nc@

selected_coefs_enet_nc <- cc_enet_min_minus_se_nc@x

names(selected_coefs_enet_nc) <- select_vars_enet_nc

# Print corresponding variables

selected_coefs_enet_nc
```

```
#>    seq1010    seq1092     seq111    seq1115    seq1178     seq121
#> -0.003884619 -0.024416859 -0.599029837 -0.008901927 -0.444230308 -0.261309599
#>    seq1277    seq1308    seq1311    seq1317    seq1330   seq1384_2
#> -0.029836243 -0.078457268  0.140429111 -0.070736997  0.167007264 -0.106924546
#>     seq147      seq17     seq243     seq248     seq257     seq291
#> -0.243111384  0.437239784 -0.081674748 -0.070349501 -0.065462794 -0.037620052
#>     seq344     seq388     seq394     seq414     seq418     seq500
#> -0.052685823  0.007682588 -0.162917113 -0.092959169  0.001974377  0.205964663
#>     seq508     seq560     seq580     seq588     seq627     seq634
#> -0.416313593 -0.343297070  0.375600849  0.218882454 -0.157964510  1.356555685
#>     seq644     seq758     seq770     seq779     seq784     seq807
#>  0.134314816 -0.109350848 -0.012346900 -0.048047259 -0.153051140 -0.004795525
#>     seq850     seq862     seq894     seq909     seq934     seq961
#>  0.161290331 -0.435320341 -0.118818292 -0.272643012 -0.275044274 -0.121455119
#>     seq992
#> -0.454803922
```

1.3 Cox lasso

```r
y <- Surv(time = train$time,
      event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
           data = train[,-(3:7), , drop = FALSE])

set.seed(1234)
cox_lasso_nc <- cv.glmnet(x = x, y = y, family = "cox",
                # family = "binomial",
                nfolds = 5,
                alpha = 0.7)

plot(cox_lasso_nc)
```




```
# Print selection
cc_lasso_min_nc <- coef(cox_lasso_nc, s = cox_lasso_nc$lambda.min)

select_vars_lasso_nc <- cc_lasso_min_nc@Dimnames[[1]][-1][cc_lasso_min_nc@i]

selected_coefs_lasso_nc <- cc_lasso_min_nc@x

names(selected_coefs_lasso_nc) <- select_vars_lasso_nc

selected_coefs_lasso_nc
```

```
#> named numeric(0)
```

```
# Model w 1SE below

cox_lambda_min_nc <- cox_lasso_nc$lambda.min
cox_lambda_min_nc_idx <- which(cox_lambda_min_nc == cox_lasso_nc$lambda)
dev_cox_lambda_nc <- cox_lasso_nc$cvm[cox_lambda_min_nc_idx]
se_cox_lambda_nc <- cox_lasso_nc$cvsd[cox_lambda_min_nc_idx]
cox_lambda_min_minus_se_idx <- which.min(abs(cox_lasso_nc$cvm - (dev_cox_lambda_nc + se_cox
cox_lambda_min_minus_se <- cox_lasso_nc$lambda[cox_lambda_min_minus_se_idx]

cc_lasso_min_minus_se_nc <- coef(cox_lasso_nc, s = cox_lambda_min_minus_se)

select_vars_lasso_nc <- cc_lasso_min_minus_se_nc@Dimnames[[1]][-1][cc_lasso_min_minus_se_

selected_coefs_lasso_nc <- cc_lasso_min_minus_se_nc@x

names(selected_coefs_lasso_nc) <- select_vars_lasso_nc

# Print corresponding variables
```

```
selected_coefs_lasso_nc
```

```
#>    seq1010    seq1092     seq111    seq1115    seq1178     seq121
#> -0.003884619 -0.024416859 -0.599029837 -0.008901927 -0.444230308 -0.261309599
#>    seq1277    seq1308    seq1311    seq1317    seq1330   seq1384_2
#> -0.029836243 -0.078457268  0.140429111 -0.070736997  0.167007264 -0.106924546
#>     seq147     seq17     seq243     seq248     seq257     seq291
#> -0.243111384  0.437239784 -0.081674748 -0.070349501 -0.065462794 -0.037620052
#>     seq344     seq388     seq394     seq414     seq418     seq500
#> -0.052685823  0.007682588 -0.162917113 -0.092959169  0.001974377  0.205964663
#>     seq508     seq560     seq580     seq588     seq627     seq634
#> -0.416313593 -0.343297070  0.375600849  0.218882454 -0.157964510  1.356555685
#>     seq644     seq758     seq770     seq779     seq784     seq807
#>  0.134314816 -0.109350848 -0.012346900 -0.048047259 -0.153051140 -0.004795525
#>     seq850     seq862     seq894     seq909     seq934     seq961
#>  0.161290331 -0.435320341 -0.118818292 -0.272643012 -0.275044274 -0.121455119
#>     seq992
#> -0.454803922
```